computing@tanet.edu.te.ua
www.tanet.edu.te.ua/computing

ISSN 1727-6209
International Scientific Journal of Computing

REMOTE LEARNING: A WORLD-WIDE-WEB OPERATED ROBOT ARM

Matanya Elchanani, Tarek M. Sobh, Raul Mihali

School of Engineering and Design, 169 University Avenue, Bridgeport, CT 06601, U.S.A.
Phone: (203) 576-4116, Fax: (203) 576-4766

Abstract: *The World-Wide-Web has been used extensively in the past few years of its existence for data exchange, and information gathering. Web online control, on the other hand, is a new emerging field, which has not yet been fully exploited and holds in it a great impact on currently available control systems. This paper discusses an application of online Internet control service - a WWW controlled robotic manipulator arm.*

Keywords: *Web / internet based control, remote manipulation, teleoperation*

1. INTRODUCTION

The WWW application is an actual real-time, online application in which the user interacts with a robot controller in real time using the World-Wide-Web as the medium. The project itself was conceived with the following ideas in mind:

1. A simple client side interface that will use only a form capable web browser.
2. User friendly interface.
3. Structural configuration that will allow fast updates for both user interface and hardware change.
4. Web interface that will allow easy implementation of distant-learning tutorials and exercises using the robot interface.

The above last requirement is quite unique in this application. A major consideration for this project was to be targeted towards distant-learning programs in the field of control and robotics, although such an interface is readily implementable in other fields as a distant-learning aid.

2 GENERAL SYSTEM DESCRIPTION

As mentioned earlier, the client side interface is comprised solely of a forms capable web-browser such as Netscape Navigator V.2 or Internet Explorer V.3.

The lack of proprietary software or hardware on the client side enables every user with web access to fully control the hardware, regardless of its platform (except of the above minimal requirements). In order to start a link with the robot, the user simply points the web browser to the robots web page. From there on, the user is guided through a series of screens that eventually end up in an input page (such as a form or a sensitive map page). When the user satisfies the input page request (by either inputting numerical data or clicking on a sensitive map image), the data is being sent as a form GET method to the server side interface.

At the server side, the user's data is processed and is being used to activate the robot arm. After the arm has ceased moving, two video cameras are being used to grab an image of the robot and feed it back to the user as an indication for the end of movement and as an opening argument for the next command. The server side interface is comprised out of three main parts that communicate with each other to accomplish the bi-directional interface between the user and the robot arm. The following are brief descriptions of the server side parts:

2.1 The web server interface

The web server interface is a series of HTML pages that are being used as the visual interface with the user. Depending on the user selection, different pages will be selected, until, eventually, a page containing an input field will be selected. When the SEND button in the page will be pressed, the page will launch a generic CGI program that will receive the data entered in the page, and according to the page that launched it will decide upon the next action. In any case, the CGI program will open an INTERNET stream socket link [1] to the ROBOT SERVER (see next part) that will be used as a data path for the flow of commands to the robot and visual feedback back to the user. No checks are being done on the commands sent to the robot server. At an early stage it was decided that most

processing and command parsing will take place on the robot server PC (which is a dedicated machine) and not on the web server machine (which is a public, general purpose machine).

2.2 The robot server

The robot server is a Pentium PC that hosts the robot server program. This program is the heart of the system and most of the processing is done here.

The PC is equipped with an RS232 interface that is used to communicate with the robot controller, an Ethernet card which is being used to communicate with the web server interface (see above), and a dual input VGA frame grabber attached to two Black & White video CCD cameras to enable visual feedback. The program is an event driven loop which senses mainly a dedicated INTERNET socket for requests (which are coming from the web server interface). The program idles as long as no requests are sensed. Whenever a request is received over the socket, the program wakes up and parses the request. The current version of the program supports two types of requests, a visual only, and a movement/visual one. A visual only request will initiate an image grabbing session that will grab the arms image using either of the cameras (depending on the request) do some processing on it (also dependent on the request - see stereoscopic interface), convert the image to a JPEG format and send it back to the web server interface. A movement/visual request will first be parsed and checked to be a legal movement command for the robots controller. It will then be passed to the inverse kinematics module [2] which will extract the new joint angles required for the new position from the 3D positional parameters given to it. The joint angles will be checked out against the current ones to validate that no limits are exceeded and finally a movement command will be sent to the controller via the RS232 interface. The main program loop will then wait on the robot to complete its movement and will initiate a visual request (see above).

2.3 The robot unit

The robot unit is comprised out of an SIR-1 semi industrial grade articulated manipulator arm with five degrees of freedom (out of which only three are being used now), and the robots controller. The controller is a CPU controlled unit that is connected to the arms motors and to a variety of position and speed sensors mounted on the arm. The controller can receive commands to move the robot from either a control panel or using a special command language through an RS232 port.

3 THE USER INTERFACE

As written earlier, the modularity of the system enables the creation of dfferent user interfaces which are using the same link to operate the robot. Currently, two user interfaces are available: the coordinate system interface (see figure 1), and the stereoscopic visual interface (see figure 2 and figure 3).

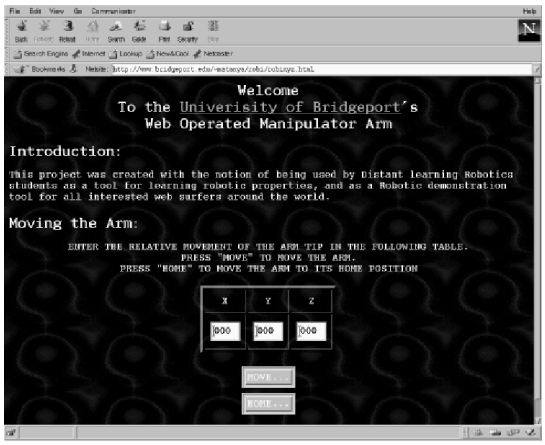

Figure 1: The coordinate system interface

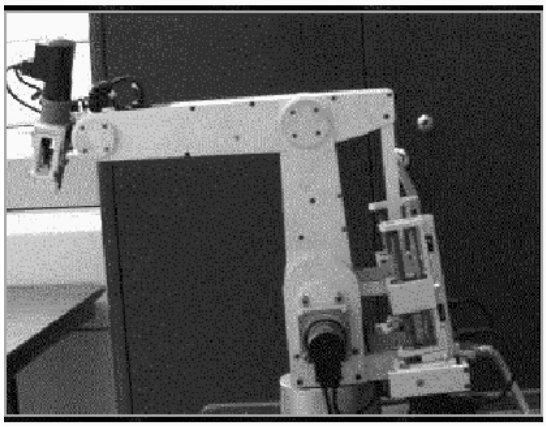

Figure 2: The stereoscopic visual interface - state 1

The coordinate system interface uses a page that requests the coordinates of a 3D point to which the robot will be moved. When the user presses the MOVE button, a movement / visual request (see above) is being sent to the robot server. After the robot has reached its final position, the new position image is being sent back to the user. A HOME button is also available which causes the robot to move to its cold-start home position. The stereoscopic interface, on the other hand, starts with a visual request at the moment the page is requested.

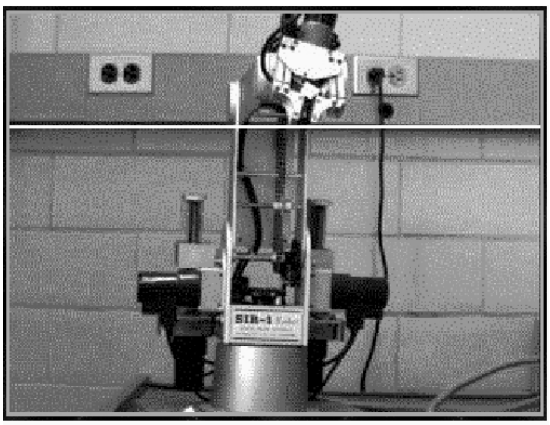

Figure 3: The stereoscopic visual interface - state 2

This enables the user to get the latest, most updated picture of the robot. After the image is grabbed from the main camera (a camera positioned in front of the arm) it is being sent as a sensitive image back to the user (figure 2). The user is then given the choice to click on the sensitive image to determine the initial X-Y coordinates of the requested position. when the user clicks on the image, the web server interface saves the coordinates entered and sends another visual request from the robot server. This time though, the image is grabbed from a second camera placed on the same horizontal plane as the main camera, but at a 90 degree angle from it. The robot server uses the Y coordinate information from the above user input to interpolate a white line across the image to mark the requested X-Y 2D point over the rotated image. The image is then sent to the web server interface which wraps it with an HTML package. It also puts hidden HTML fields into which the previously selected X-Y coordinates are placed, so they will be passed with the final user request. Eventually, the user receives another image of the robot which is the 90 degree side view with a white line, representing his previous selection (figure 3). The user is requested to click again over the image, thus, setting the new images X coordinate. This coordinate is basically the Z coordinate (X system, rotated 90 to be perpendicular to the X-Y plane). The request received contains the Z coordinate, together with the X-Y coordinates (embedded as hidden HTML fields), and is being sent as a movement/visual request to the robot server. The robot server is using this data to move the robot to the new position and finally send the resultant picture back to the user.

4 FUTURE ADDITIONS

The WWW operated arm is a dynamic on-going project. Currently, there are a few additions that are to be considered to be added to the system:

- Queuing system: During the writing of this paper, extensive work is being done in creating a queuing system for user control and authentication. For the short duration of its life, the WWW robot had received thousands of hits and an urgent need for a queuing system was noticed. The queuing system we are designing relies solely on a state machine comprised out of the user's own web pages. Each viewed page contains a special magic cookie (inside a hidden field) which is decrypted in the server and carries information on scheduling, expiration, priority and user level.
- Java based interface: A Java based interface is also considered as a future addition. The interface would help to move some of the complex inverse kinematics calculations from the robot server to the user's machine.
- Live Video: For wideband Internet connections, a live video frames stream of the robot's movement would enhance the understanding of what is actually happening on the robot side. We are still considering either a server push, or a Java video console. With both of the considered options, our prime directive is a non proprietary client side software.
- More Distant Learning: Expending the project into the general sciences. For example, using the robot and interface in a distant learning chemistry lab experiment set (we are actually working now on this one)

5 RELATED INFORMATION

Information about this project and other robotics related projects is provided on the University of Bridgeport's Robotics and Intelligent Sensing and Control laboratory web page at:

http://www.bridgeport.edu/sed/risc

Feel free to visit us at the above page, or go directly to the WWW robot arm page at:

http://www.bridgeport.edu/~matanya/robi

(please remember that as this project is in its early stages of development, it is mostly online and under constant module and interface change).

6 CONCLUSION

While the project is an ongoing work and it has proven to be a practical foundation for many future directions of enhancement, it has clearly shown the large interest towards remote control, tele manipulation, web based interaction as well as distance learning.

At the time this project has been tested, the existence of similar projects over the Internet was minimal and in fact we were not able to refer to comparable systems for the purpose of a vaster reference.

REFERENCES

[1] Stevens, W. Richard. Unix Network Programming, Prentice Hall, Englewood Cli_s, N.J., 1990.

[2] Spong, W. Mark, Robot Dynamics and Control, John Wiley & Sons, N.Y., 1989.

Tarek M. Sobh *received the B.Sc. in Engineering degree with honors in Computer Science and Automatic Control from the Faculty of Engineering, Alexandria University, Egypt in 1988, and M.S. and Ph.D. degrees in Computer and Information Science from the School of Engineering, University of Pennsylvania in 1989 and 1991, respectively. He is currently the Director of the School of Engineering and Design; director of external engineering programs; Professor of Computer Science and Computer Engineering; the founding director of the interdisciplinary Robotics, Intelligent Sensing, and Control (RISC) laboratory at the School of Engineering and Design, University of Bridgeport, Connecticut; and the Chairman of the Discrete Event Dynamic Systems (DEDS) Technical Committee of the IEEE Robotics and Automation Society.*

He was the Interim Chairman of Computer Science and Engineering at the University of Bridgeport in 1998 and was a Research Assistant Professor of Computer Science at the Department of Computer Science, University of Utah from 1992 -- 1995, and a Research Fellow at the General Robotics and Active Sensory Perception (GRASP) Laboratory of the University of Pennsylvania during 1989 – 1991. Dr. Sobh's current research interests include reverse engineering and industrial inspection, CAD/CAM and active sensing under uncertainty, robots and electromechanical systems prototyping, sensor-based distributed control schemes, unifying tolerances across sensing, design, and manufacturing, hybrid and discrete event control, modeling, and applications, and mobile robotic manipulation.

He has published over 80 journal and conference papers, and book chapters in these and other areas. Dr. Sobh is a Licensed Professional Electrical Engineer (P.E.), a Certified Manufacturing Engineer (CMfgE) by the Society of Manufacturing Engineers, and a Certified Reliability Engineer (C.R.E.) by the American Society for Quality, a member of Tau Beta Pi, Sigma Xi, Phi Beta Delta, and Upsilon Pi Epsilon.

Dr. Sobh was the recipient of the Best Paper Award at the World Automation Congress Conference (WAC 98). Dr. Sobh is a member or senior member of several professional organizations including; ACM, IEEE, IEEE Computer Society, IEEE Robotics and Automation Society, the National Society of Professional Engineers (NSPE),the American Society of Engineering Education (ASEE), and the Society of Manufacturing Engineers (SME).

Raul C. Mihali *received the B.Sc. and M.S. degree in Computer Programming degree with honors in Computer Science and Automatic Control from Department of Computer Science and Engineering, University of Bridgeport, CT, USA, in 1999 and 2000. He is currently completing the Ph.D. degree in Computer Engineering at University of Bridgeport.*

His current research interests include CAD/CAM and active sensing under uncertainty, robots and electromechanical systems prototyping, modeling and applications, mobile robotic manipulation, algorithms and programming techniques, stealth and security oriented robots, polymorphic structures and robot actuators and sensors for unusual execution tasks.